ConGRets: Contrastive Learning for Scalable, Low-latency Co-speech Gesture Generation with Zero-shot Speaker Style Adaptation

ANONYMOUS AUTHOR(S)
SUBMISSION ID: PAPERS-2545

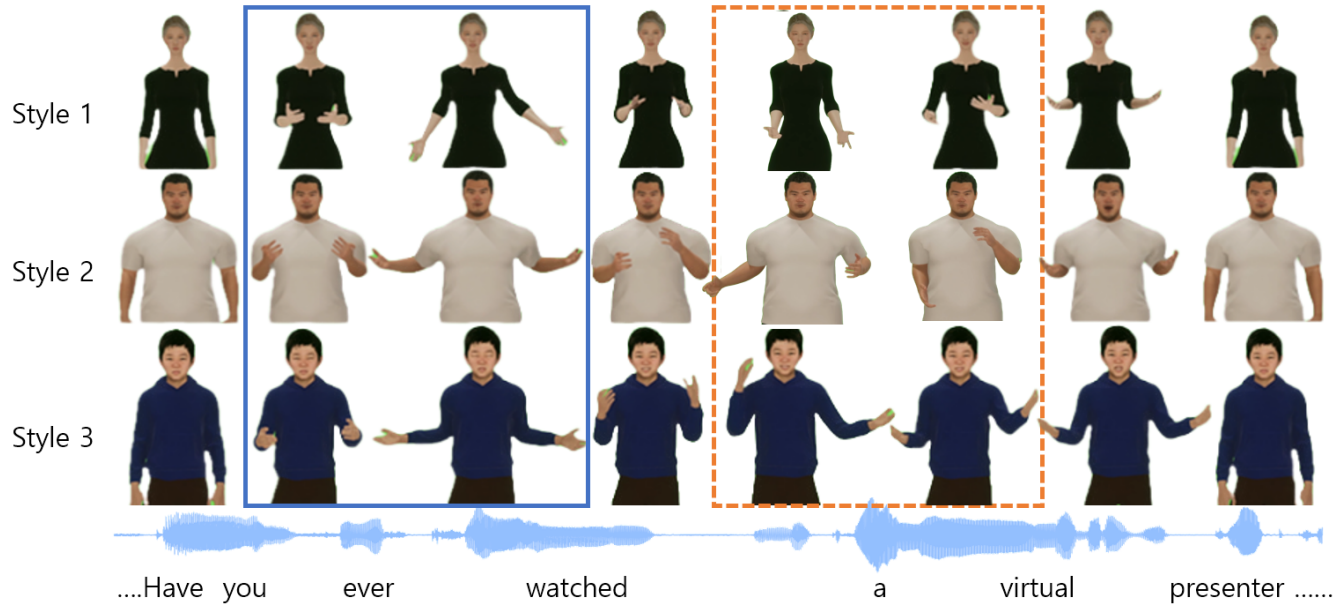


Fig. 1. ConGRets generates co-speech gestures across three distinct speaker styles, maintaining alignment with both speaker-specific motion patterns and speech content. Solid-line box indicate identical gestures rendered in different styles, while dotted-line box depict varying gestures corresponding to the same speech content.

This paper presents **ConGRets**, a scalable, low-latency framework for co-speech gesture generation that supports zero-shot speaker personalization without requiring audio or speaker identity labels. Unlike conventional end-to-end approaches, which either synthesize gestures frame-by-frame or discretize motion into tokens before decoding, ConGRets retrieves pre-processed gesture units directly from a curated library using a contrastive learning-based retrieval mechanism. Input text and a motion-derived speaker style vector are projected into a shared latent space, enabling accurate and expressive gesture selection without the overhead of generative decoding. A key contribution of ConGRets is its **text-only design**, which allows seamless integration with real-time chatbot and TTS pipelines without introducing dependencies on audio processing or prosodic alignment. Although the system operates on compact gesture units, it maintains fluidity across utterances by employing overlap-aware segmentation and runtime blending, ensuring coherent motion transitions without sequence-level prediction. We evaluate ConGRets across multiple dimensions, including retrieval accuracy, gesture quality, execution speed, and speaker style consistency, and benchmark against rule-based, generative, and zero-shot baselines. Results show that ConGRets delivers perceptually natural gestures, robust zero-shot adaptation, and superior real-time performance, making it a compelling alternative to end-to-end generation for expressive digital humans and conversational agents.

CCS Concepts: • **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Additional Key Words and Phrases: Do, Not, Us, This, Code, Put, the, Correct, Terms, for, Your, Paper

ACM Reference Format:
Anonymous Author(s). 2018. ConGRets: Contrastive Learning for Scalable, Low-latency Co-speech Gesture Generation with Zero-shot Speaker Style Adaptation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Synchronized body movements accompanying speech, commonly known as co-speech gestures, are fundamental to human communication. These gestures enhance the expressiveness and realism of digital humans, making them essential in Human-Computer Interaction (HCI) applications such as conversational agents, educational avatars, and immersive AR/VR environments [32, 45, 46, 52].

Despite significant progress, generating speaker-specific co-speech gestures in real time remains a challenging task. Existing approaches span rule-based systems, data-driven mappings, and deep learning-based generative models. Rule-based and data-driven methods are often responsive and efficient [12, 32, 38, 42], but typically suffer from limited gesture diversity, weak generalization across speakers, and difficulty adapting to new contexts. On the other hand, generative models synthesize gestures directly from text or speech input [3, 21, 50, 51], addressing some of these issues but introducing new challenges, such as high inference cost, gesture averaging, and slow response time, particularly problematic in real-time applications [43].

To manage the many-to-many nature of gesture expression, many recent generative systems introduce intermediate motion representations—such as gesture tokens [7, 54] or lexicons [6], that are decoded into motion via learned decoders. While these representations increase diversity and control, they also introduce decoding latency and reconstruction artifacts. This leads to a practical question: if gesture generation ultimately selects from a finite set of discrete motions, could a retrieval-based approach better support low-latency interaction?

Prior work has explored retrieval paradigms to address this. One early system proposed a rule-based framework that remains widely adopted in platforms like the Virtual Human Toolkit [22, 24, 38]. Building on this, subsequent efforts constructed automatic text-driven rule-based systems using curated gesture libraries and TED-style transcripts [51], supporting fast prototyping and real-time demo deployment in interactive environments [25–29]. These methods exhibited low latency for small-scale inputs but faced scalability issues due to large memory footprints and brute-force text-gesture matching, limiting their practicality for batch processing or large dialogue contexts.

Recent research has further demonstrated the effectiveness of retrieval frameworks when combined with learning-based representations. For example, graph-based unit selection has shown improved realism [55], and contrastive learning has been used to enhance gesture relevance and style consistency across languages and motion domains [5]. However, rule-based systems remain constrained by high memory demands and slow large-scale retrieval, and generative methods continue to struggle with inference speed and diversity trade-offs.

Moreover, empirical findings show that virtual agent co-presence and user engagement are not significantly degraded by using TTS over real voice, assuming sufficient audio quality [1]. Likewise, text-based gesture systems often perform comparably to their speech-based counterparts in generating appropriate motion [8]. These observations support the use of text-only inputs as a practical design choice for scalable, modular pipelines.

We introduce ConGRets, a contrastive learning-based gesture retrieval framework designed for low-latency, scalable, and style-consistent co-speech gesture generation. Instead of decoding gestures from latent vectors, ConGRets retrieves them directly from a curated motion library using contrastively learned representations of text and motion style. This approach leverages the strengths of retrieval, efficiency, reusability, and robustness, while avoiding the overhead of decoding or retargeting.

ConGRets consists of two main components: (1) a contrastively trained speaker style encoder that captures stylistic traits purely from motion, without requiring speaker identity or audio; and (2) a compact BERT-based text encoder, fused with the style embedding and aligned to a frozen motion encoder (GestureCLR [4]) through contrastive learning. During inference, fused representations are matched to cluster centroids in a pre-segmented gesture unit library.

The gesture library comprises 2–3 second segments extracted from high-variance motion windows, selected using variance thresholds and boundary similarity constraints [5]. Gesture units are clustered per speaker using bisecting K-Means, enabling consistent retrieval that respects individual motion styles. While just one minute of motion is sufficient for embedding speaker-specific traits, embeddings improve with longer recordings, 10 minutes being sufficient for stable representations, and up to 60 minutes yielding broader gesture coverage.

ConGRets is fully text-driven, making it compatible with systems where text is generated or available before synthesized speech. It supports chunked inference and blends gesture units using overlap smoothing and pose constraints, ensuring visual continuity even with short segments. The system achieves sub-500ms inference for 1000 text chunks on standard hardware, without requiring any neural decoder, server-side inference, or audio processing.

Our contributions are:

(1) **A contrastively trained, zero-shot speaker style encoder** that learns motion-specific stylistic variation without labels, identity, or audio.
(2) **A latent space retrieval mechanism** aligning fused text-style vectors to gesture clusters, enabling fast and expressive retrieval without decoding.
(3) **A lightweight and scalable framework** that supports real-time use with sub-500ms latency, even for large batches, without sacrificing style or relevance.
(4) **Extensive evaluation** demonstrating that ConGRets achieves strong performance in gesture appropriateness, speaker style consistency, and Fréchet Gesture Distance (FGD), with high retrieval accuracy in zero-shot conditions.

Rather than replacing generative models, ConGRets offers a practical and extensible alternative for settings where gesture generation enhances user experience but must meet real-time and resource constraints. It is particularly suited for virtual humans in interactive applications such as AR/VR, conversational agents, and embodied interfaces.

The remainder of this paper is organized as follows: Section 2 reviews related work, Section 3 describes the methodology, Section 4

presents the dataset and experimental setup, Section 5 reports results, Section 6 discusses implications and limitations, and Section 7 concludes.

2 Related Work

2.1 Rule-Based Gesture Generation

Rule-based systems for co-speech gesture generation have long been foundational in Human-Computer Interaction (HCI) [10, 11, 13, 32, 39, 41, 43, 44]. These systems map input speech or text to predefined gestures through handcrafted rules. Notable systems include the BEAT toolkit [13], which schedules gestures based on linguistic analysis, and SmartBody [41], which supports animation synthesis via annotated gesture units. While these frameworks generate contextually appropriate gestures, they suffer from limited scalability, high manual overhead, and lack of speaker adaptability [30, 42, 48].

To address these shortcomings, retrieval-based extensions emerged. Ferstl et al. [18] introduced a system that retrieves gesture units based on prosodic and semantic matching. Habibie et al. [23] used motion-matching techniques to retrieve controllable gesture sequences. However, many of these systems still rely on audio cues or hand-designed rules. ConGRets builds upon this lineage by learning to retrieve gestures directly from text in a style-consistent latent space, eliminating the need for rule authoring or acoustic input.

2.2 Learning-Based Gesture Generation

Learning-based methods train models to synthesize gestures from multimodal inputs [8, 9, 15, 31, 34, 49, 51]. Early deep models used regression from audio to joint positions, but generated average, lifeless motions [21]. Later works introduced adversarial losses [17], probabilistic models [33], and variational embeddings [50] to improve realism and diversity.

Recent approaches integrate powerful sequence models and emphasize semantics. Kucherenko et al. [35] and Bhattacharya et al. [9] fused text and audio inputs, while Gao et al. [19] used ChatGPT to derive semantic gesture scripts from text. Diffusion models have also shown promise: Zhu et al. [56] introduced a diffusion-based model that generates fluid gestures from audio. In contrast, ConGRets emphasizes a text-only design, demonstrating that retrieval-based gesture generation guided by semantic embeddings is both efficient and expressive, even without prosodic input.

2.3 Speaker Style Adaptation in Gesture Systems

Speaker style adaptation is critical for realistic gesture generation. Early systems trained speaker-specific models [21] or used speaker IDs as conditioning [50]. Ahuja et al. [3] proposed style embeddings for transfer between speakers using multimodal signals.

More recent works enable zero- or few-shot adaptation. Fares et al. [16] developed a speaker-style encoder using audio, text, and pose to achieve zero-shot adaptation. However, reliance on audio limits applicability for text-driven systems. Ghorbani et al. [20] introduced ZeroEGGS, enabling example-based adaptation without retraining. Ahuja et al. [2] showed that gesture models can be quickly adapted to new speakers with minimal motion data.

ConGRets introduces a contrastive speaker encoder trained only on motion data. It enables zero-shot personalization by projecting both text and motion into a shared space and retrieving matching gesture units. Unlike prior methods that rely on fine-tuning or audio, ConGRets adapts styles through retrieval from curated motion units, offering scalable and dynamic personalization.

3 Methodology

3.1 System Overview

ConGRets is a contrastive learning-based, retrieval-oriented system for co-speech gesture generation with zero-shot speaker style adaptation. It decouples gesture generation from gesture selection by projecting both input text and speaker motion style into a shared latent space, then retrieving the best-matching gesture unit from a pre-clustered library.

The framework is composed of three major modules:

- **GestureCLR[4]** – a frozen motion encoder trained via contrastive learning to embed gesture units into a 10-dimensional latent space.
- **Speaker Style Encoder** – a transformer-based model that encodes speaker-specific motion style into a 768D vector.
- **Text Encoder (BERT)** – a 4-layer, 256D BERT-mini model that encodes text chunks into embeddings fused with style for gesture retrieval.

3.2 GestureCLR: Latent Gesture Representation

GestureCLR [4], inspired from the SimCLR framework [14], was trained on motion capture data from the Talking With Hands dataset [37]. Raw mocap sequences were segmented into 2–3 second gesture units using criteria established in prior work on multilingual co-speech synthesis[5]:

- Unit length must be between 2 to 3 seconds
- Motion variance must exceed a dataset-specific empirical threshold
- First and last frames must be similar in pose (loop-closure)

Gesture units are encoded into a 10D latent space. During preprocessing, 1000–2000 units per speaker are clustered into 100 balanced groups using bisecting K-Means. These clusters are fixed post-training, and only cluster centroids are used for matching at inference.

3.3 Speaker Style Encoder (Figure 7)

3.3.1 Architecture. The speaker style encoder learns to capture inter-speaker variation in motion style by hierarchically processing 3D body motion using a two-stage Transformer-based architecture. Input motion is organized as a 4D tensor:

$$\mathbf{X} \in \mathbb{R}^{B \times N \times F \times J}$$

where:

- B is the batch size (number of speakers),
- $N = 10$ is the number of non-overlapping 1-minute sub-segments per speaker,
- $F = 900$ is the number of frames per sub-segment (sampled at 15 FPS),
- $J = 150$ represents the 3D coordinates of 50 joints.

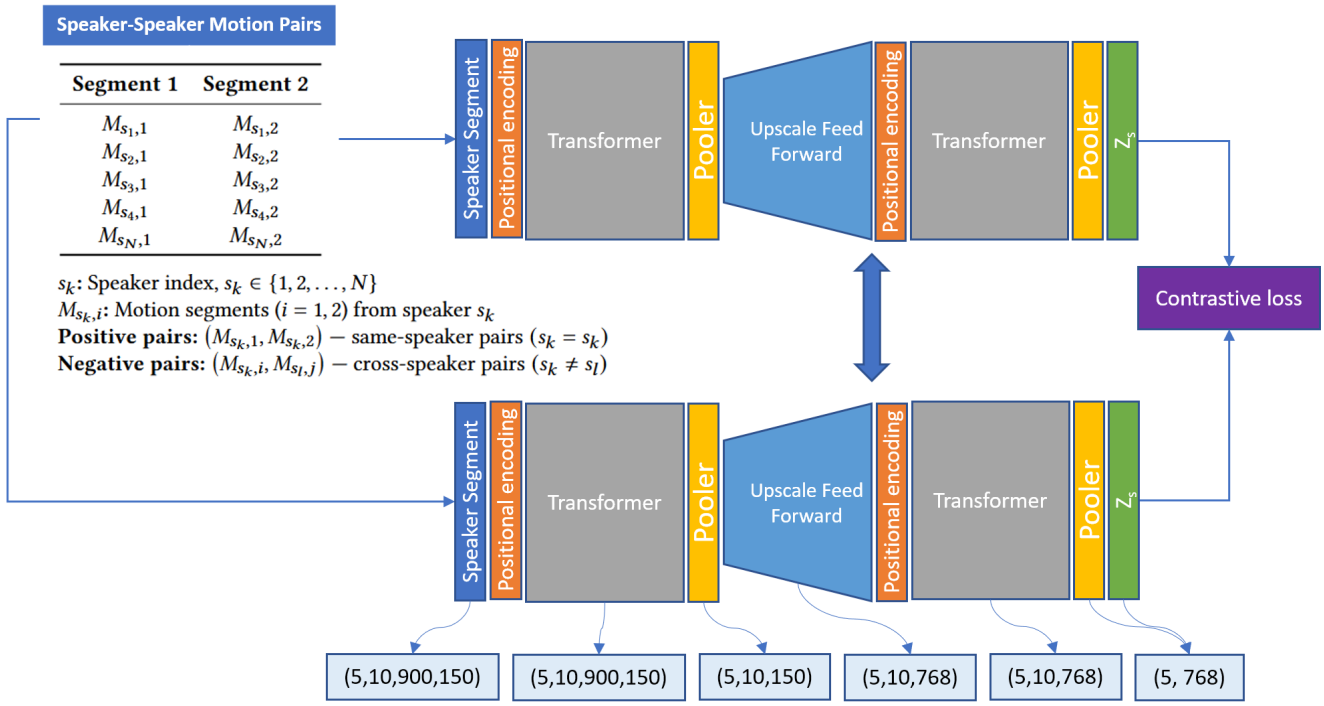


Fig. 2. Speaker Style encoder model. Example of speaker-same motion segment pairs used for contrastive training. Each row contains two non-overlapping motion segments from the same speaker. Negative pairs are formed by pairing segments from different speakers.

This setup reflects a 10-minute clip split into $N = 10$ chunks, allowing the model to first extract low-level spatial cues from individual frames and then reason over longer temporal context to encode high-level speaker motion patterns.

Stage 1: Spatial Encoding (Per Frame). Each 1-minute segment $\mathbf{X}_{b,n} \in \mathbb{R}^{F\times J}$ is first passed through a spatial Transformer encoder $\mathcal{T}_{\text{spatial}}$:

$$\mathbf{H}_{b,n} = \mathcal{T}_{\text{spatial}}(\text{PE}(\mathbf{X}_{b,n})) \in \mathbb{R}^{F\times J}$$

where $\text{PE}(\cdot)$ denotes positional encoding applied along the joint dimension. The output is then mean-pooled across frames:

$$\tilde{\mathbf{H}}_{b,n} = \frac{1}{F}\sum_{f=1}^{F} \mathbf{H}_{b,n,f} \in \mathbb{R}^{J}$$

forming a compact representation of motion for each 1-minute chunk. Stacking over N yields:

$$\tilde{\mathbf{H}}_b = [\tilde{\mathbf{H}}_{b,1}, \ldots, \tilde{\mathbf{H}}_{b,N}] \in \mathbb{R}^{N\times J}$$

Stage 2: Temporal Encoding (Across Segments). These per-segment vectors are projected to 768D via a feedforward layer:

$$\mathbf{U}_b = \phi(\tilde{\mathbf{H}}_b) \in \mathbb{R}^{N\times 768}$$

where ϕ is a learned projection. The result is fed to a temporal Transformer encoder $\mathcal{T}_{\text{temporal}}$:

$$\mathbf{Z}_b = \mathcal{T}_{\text{temporal}}(\text{PE}(\mathbf{U}_b)) \in \mathbb{R}^{N\times 768}$$

and temporally pooled to obtain the final speaker style embedding:

$$Z_s^{(b)} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{Z}_{b,n} \in \mathbb{R}^{768}$$

This hierarchical encoding ensures that the model first captures localized joint dynamics (low-level motion) and then aggregates temporal consistency and higher-level behavioral traits over several minutes of data. It also enforces a minimum input requirement of 1 minute of motion (i.e., at least one $F = 900$ frame segment) for style inference.

3.3.2 Training. We train the encoder using the NT-Xent contrastive loss [14], which encourages embeddings of motion segments from the same speaker to cluster together, while pushing apart those from different speakers.

For a batch of B speakers, each providing two disjoint 10-minute clips $\mathbf{X}_a, \mathbf{X}_b$, the model outputs two style embeddings $Z_s^{(a)}, Z_s^{(b)} \in \mathbb{R}^{768}$ per speaker. The loss for a positive pair (i, j) is given by:

$$\mathcal{L}_{i,j} = -\log \frac{\exp\left(\text{sim}(Z_i, Z_j)/\tau\right)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} \exp\left(\text{sim}(Z_i, Z_k)/\tau\right)}$$

where $\text{sim}(u, v) = \frac{u^\top v}{\|u\|\|v\|}$ is cosine similarity and τ is the temperature parameter.

The model is trained using segment pairs from the same speaker as positives and all other pairs as negatives. We find that even 10

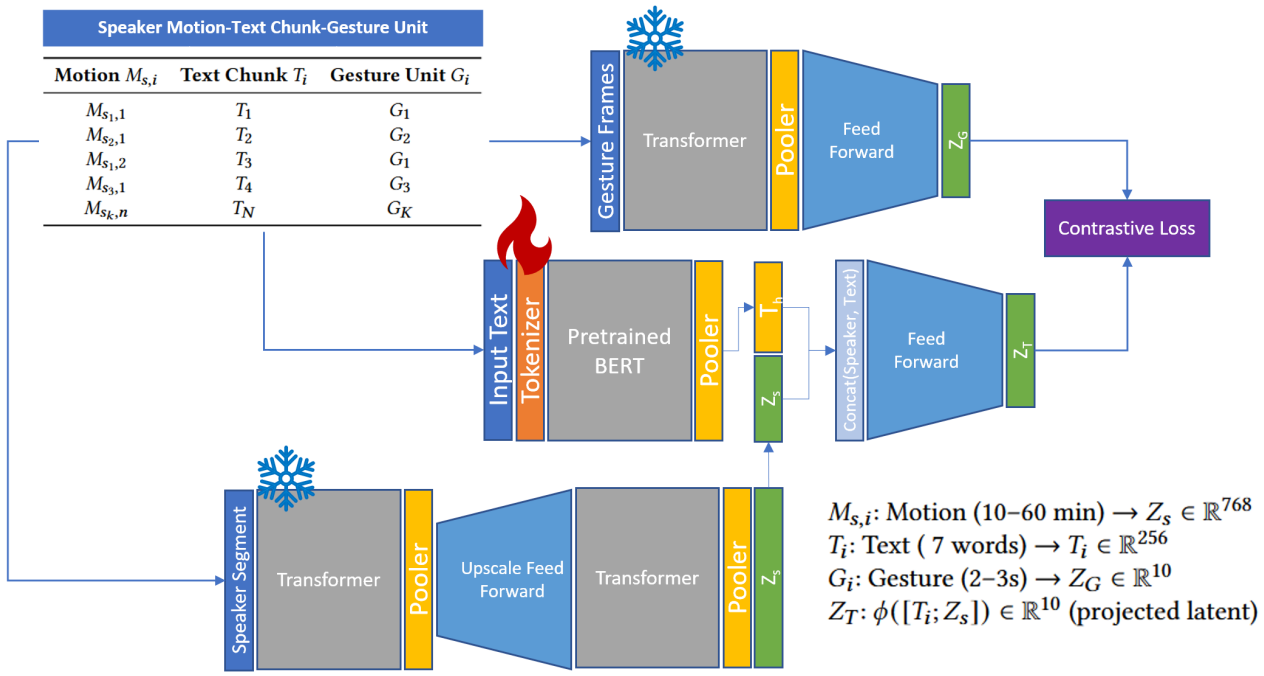


Fig. 3. ConGRets with Speaker Style encoder. Speaker Motion–Text Chunk–Gesture Unit training triplets used for contrastive learning. Each row includes a motion segment used to compute the speaker style latent Z_s via frozen speaker style enocder, a corresponding text chunk T_i, and a gesture unit G_i encoded via frozen GestureCLR.

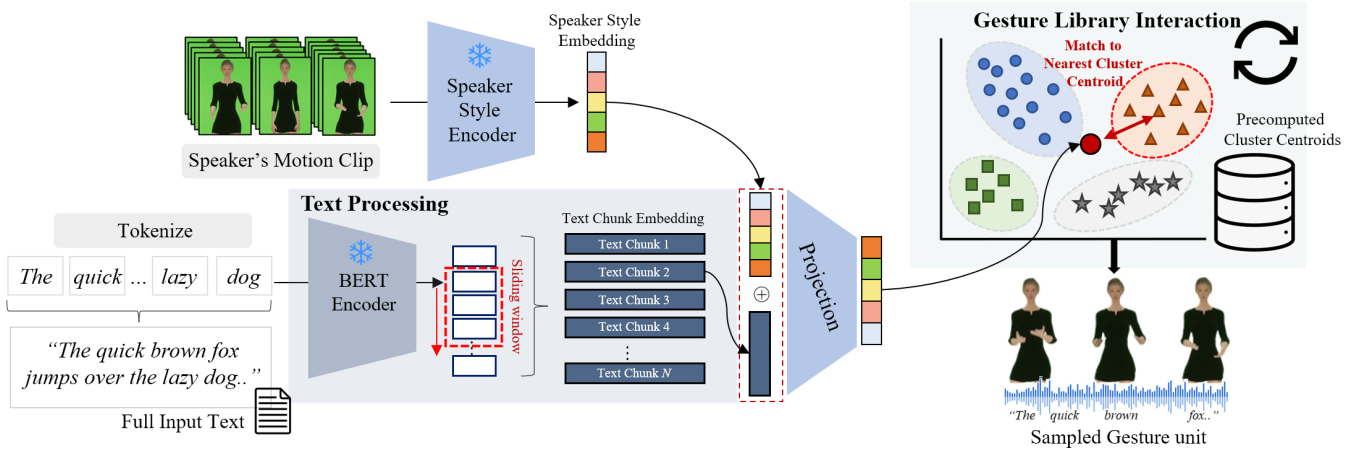


Fig. 4. ConGRets inference pipeline

minutes of motion data suffices to learn a meaningful and discriminative speaker style vector, though performance and generalization improve with longer recordings (e.g., 60 minutes per speaker).

3.4 Text Encoder and Fusion (Figure 3)

ConGRets fuses semantic and stylistic information to retrieve gesture units from a shared latent space. The architecture processes triplets: text chunk T_i, gesture unit G_i, and a motion segment $M_{s,i}$ used to compute the speaker style.

Text Encoding: Each text chunk T_i (approx. 7 words) is passed through a 4-layer BERT-mini model (output: 256D). Token embeddings $X \in \mathbb{R}^{n\times 256}$ are pooled via attention-masked mean:

$$T_i = \frac{\sum_{j=1}^{n} X_j \cdot M_j}{\sum_{j=1}^{n} M_j}, \quad T_i \in \mathbb{R}^{256}$$

Style Conditioning: A motion segment $M_{s,i}$ (10–60 minutes) is passed through the **frozen** speaker style encoder (see Figure 2) to produce a style vector:

$$Z_s \in \mathbb{R}^{768}$$

Fusion and Projection: Text and style embeddings are concatenated:

$$[T_i; Z_s] \in \mathbb{R}^{1024}$$

The fused vector is passed through a feedforward layer ϕ to project it into the shared 10D latent space:

$$Z_T = \phi([T_i; Z_s]) \in \mathbb{R}^{10}$$

Only the text encoder and the projection layer ϕ are trained. The speaker style encoder and GestureCLR model are kept frozen throughout.

Contrastive Training Objective. The goal is to align the fused latent vector Z_T with the corresponding gesture latent Z_G (from the frozen GestureCLR model). Gesture units are pre-encoded into the 10D space and not updated during training.

We apply the same NT-Xent loss defined in the speaker style encoder section, treating Z_T as the query and Z_G as the target. Positive pairs come from aligned (text, gesture) samples, and all other combinations are negatives.

This setup enables the model to learn content- and style-aware gesture retrieval using only frozen motion representations and contrastive supervision.

3.5 Gesture Retrieval at Inference

At inference, gesture retrieval proceeds as follows:

(1) A 10-minute motion clip is used to compute the speaker's style embedding via the pre-trained style encoder.
(2) The input text is divided into segments of approximately seven words.
(3) Each segment is encoded using a BERT model, concatenated with the style vector, and projected into the latent gesture space.
(4) The resulting latent vector is matched to precomputed cluster centroids representing gesture units (clustered per speaker).
(5) A gesture unit is sampled from the nearest cluster for playback.

This process is repeated for each segment. Chunking introduces the risk of splitting sentences in semantically awkward ways, which can lead to degraded matching quality. To address this, we use a bidirectional BERT encoder and avoid hard splits. Instead, the full sentence is tokenized and processed as a whole, and a sliding window approach is applied to extract embeddings for each seven-word chunk. Pooling is performed using only the tokens corresponding to the target chunk, ensuring local semantic focus while retaining contextual richness.

If a segment contains fewer than seven words (e.g., at sentence boundaries), it is passed directly. Additional adjustments are made for very short final segments (less than three words) to avoid unstable representations.

Gesture units are assumed to be pre-cleaned during preprocessing. During deployment, any unit found to be faulty, such as those with pose artifacts or poor retargeting, is flagged at runtime and excluded from future retrieval via a filtering mechanism.

3.6 Ablation: Gesture Decoding

To validate that generation quality is a function of decoder capacity, we trained a VAE-style decoder that reconstructs gestures from GestureCLR latent vectors. The decoder is a mirrored version of the GestureCLR encoder and demonstrates that low FGD scores in generated output stem from decoder limitations not the latent representation itself. This supports our design choice of decoupling generation from retrieval.

4 Dataset and Experimental Setup

This section outlines the dataset, preprocessing pipeline, model training setup, and evaluation protocols used to assess the performance of the ConGRets framework.

4.1 Dataset: BEAT Corpus and Rulemaps

We use the BEAT dataset [40], a large-scale motion-captured corpus comprising recordings from 30 speakers. For training and evaluation, we employed a 20-5-5 speaker split into training, validation, and test sets. Multiple random splits were tested, and the final model was retrained using the best-performing configuration, which included the original 5 validation speakers. The held-out test set was reserved for final evaluation only.

To facilitate text-gesture alignment, we adopt the rulemap-based segmentation strategy from Ali et al. [5]. Each speaker's recordings were segmented into gesture units (2–3 seconds) and paired with corresponding text based on timestamp alignment. Rulemaps for all 30 speakers were created using this method, ensuring consistent alignment. Gesture units were filtered using three criteria: duration (2–3s), motion variance (above a dataset-specific empirical threshold), and frame closure (first and last frames should be similar). Each speaker contributed between 1000 and 2500 gesture units.

4.2 Gesture Unit Clustering

During training, ConGRets was provided with full text–gesture unit pairs to maximize diversity. Gesture units were clustered only at inference time. We applied bisecting K-Means clustering per speaker to organize 1500–2500 gesture units into 100 balanced clusters. At runtime, retrieval is performed by matching the latent output to cluster centroids and sampling from the best-matching cluster.

4.3 Model Training Configuration

Text Encoder and ConGRets Training: We used a 4-layer BERT-mini model (256D output) from Google's pretrained repository [47] and fine-tuned it on our dataset. The BERT output was concatenated with a 768D speaker style vector (from the style encoder), producing a 1024D embedding projected to a 10D latent vector. Training used

the NT-Xent contrastive loss to align the ConGRets output with latent representations from GestureCLR. Batch sizes were 128 (train) and 32 (validation), with a maximum of 1000 epochs and early stopping (patience = 10). AdamW was used with a learning rate of 5×10^{-4} and cosine annealing schedule.

Speaker Style Encoder: The encoder was trained using contrastive learning with positive pairs sampled from non-overlapping motion clips of the same speaker. Motion sequences were independently encoded via spatial and temporal Transformer blocks. Cosine similarity and NT-Xent loss were used to encourage speaker-specific clustering in the 768D style space.

4.4 Evaluation Setup

We evaluated ConGRets along four axes:

- **Execution Time:** Median gesture retrieval latency (in milliseconds) was measured per batch (1000 samples) on both CPU and GPU systems. Comparisons were made with Ginosar et al. [21], Yoon et al. [50], and a rule-based system based on Ali et al. [5].
- **Speaker Style Adaptability:** We computed cosine similarity between ConGRets outputs and the speaker's own motion cluster centroid. The baseline for comparison was the model proposed by Fares et al. [16], retrained per their hyperparameters and setup.
- **Retrieval Accuracy:** Using latent outputs from ConGRets and frames output from Fares et al.[16] encoded through GestureCLR to obtain latents, we computed accuracy based on the gesture cluster nearest to the original latent.
- **Gesture Quality:** We used Fréchet Gesture Distance (FGD) scores from the BEAT dataset benchmark, comparing against baseline models including Seq2Seq[51], DiffuGestureCLIP[7], CaMN[40] and Semantic Gesticulator[54]. ConGRets was evaluated with the speaker style encoder and against a VAE-based gesture decoder trained as an ablation to test the quality of latent gesture representations.

In all comparisons, inference was performed on held-out test speakers. More details and quantitative results are provided in the Results section.

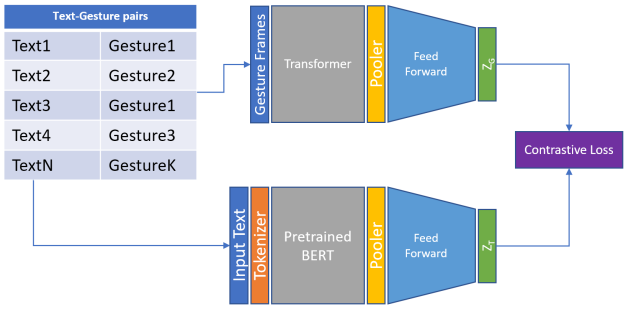


Fig. 5. ConGRets without Speaker Style encoder

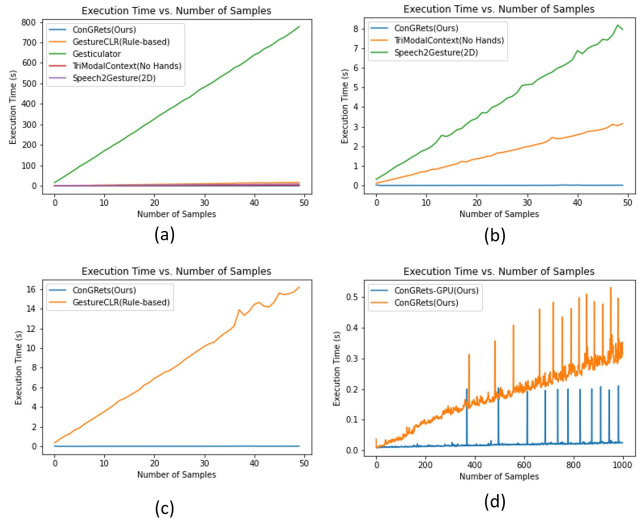


Fig. 6. ConGRets execution time as compared to baseline

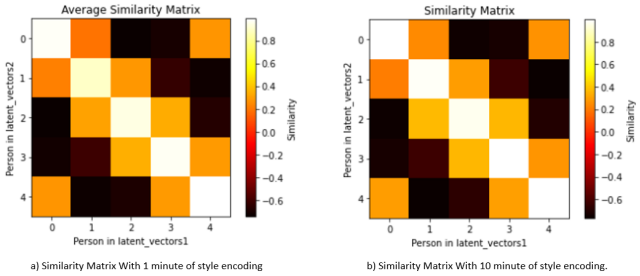


Fig. 7. Speaker Style encoder matrix

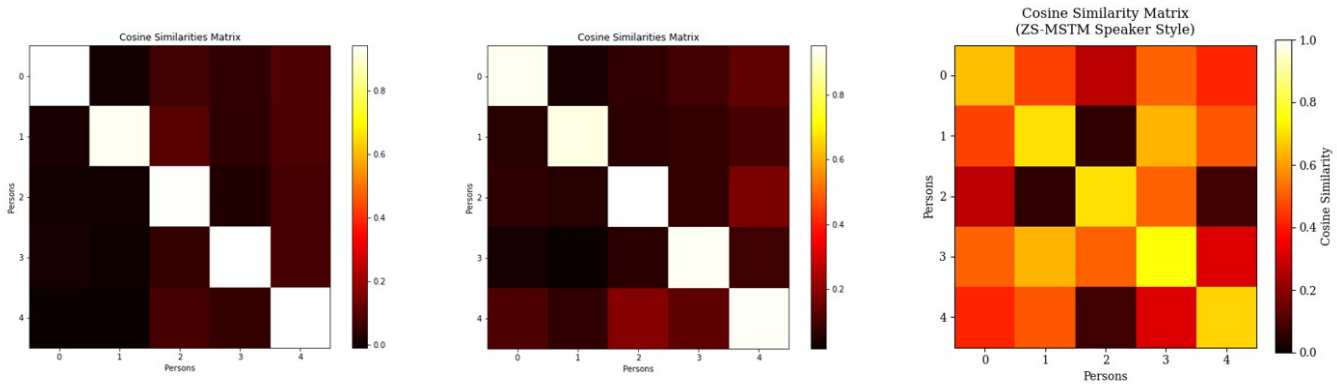
5 Results

We evaluate ConGRets across four axes: execution speed and memory efficiency, retrieval performance, zero-shot speaker adaptation, and gesture quality (via FGD). Comparisons are made against prior rule-based and end-to-end gesture generation systems.

5.1 Speed and Memory Efficiency

Although gesture generation research frequently focuses on output quality, execution time and memory usage are often underreported—despite being critical for real-time and embedded applications. Among recent systems, GestureDiffuCLIP [7] is one of the few to explicitly report runtime, requiring 1.5 seconds to generate a 1-second (60-frame) gesture sequence on a Tesla V100 GPU. However, this level of resource usage is often impractical for large-scale, interactive environments or lightweight deployment.

To provide a clearer picture of runtime behavior, we benchmarked several gesture generation systems with publicly available code, prioritizing those with efficient architectures. Table 2 presents execution time (on CPU), memory usage, and gesture quality metrics for these baselines, along with ablations of our proposed method.



a) Similarity matrix with separate model for each speaker

b) Similarity matrix with single model speaker style encoding

c) Similarity matrix of ZS-MSTM speaker style encoding

Fig. 8. Speaker Style adaptation matrix

Table 1. Retrieval accuracy% as a function of context length. Percentages are rounded off. Abbreviations: SP=Speaker-Style; CG=Content-Gesture; CC=Content-Cluster.

Style-Context	ZS-MSTM↑			ConGRets↑			Encoded ConGRets↑		
	SP	CG	CC	SP	CG	CC	SP	CG	CC
5 min	35	11	32	82	15	54	54	10	26
10 min	43	16	44	94	24	68	61	12	33
20 min	57	21	49	96	35	81	65	22	39
30 min	71	45	64	99	55	89	76	26	42

ConGRets achieves significantly better runtime and memory efficiency while maintaining high output quality. It processes 1000 text queries in under 500ms on CPU (i.e., 0.5ms per query), and requires less than 100MB of memory. This contrasts sharply with the rule-based system, which takes 16 seconds for just 50 queries and exceeds 2GB of memory usage, primarily due to linear search and large pre-trained text embeddings. Even TrimodalContext [50], a relatively efficient deep model, requires 3 seconds for 50 queries, while Speech2Gesture [21] takes 8 seconds for the same batch size.

Figure 6(c,d) further illustrates the scaling characteristics: ConGRets retains low-latency responsiveness even with large batches, due to its lightweight architecture and cluster-based indexing scheme. These results underscore the practical viability of ConGRets for deployment in resource-constrained or real-time systems, where both speed and expressiveness are essential.

5.2 Gesture Quality via Fréchet Gesture Distance (FGD)

Table 2 reports Fréchet Gesture Distance (FGD)[50] scores evaluated on the BEAT dataset. FGD measures the distance between the distribution of generated gestures and real motion, with lower scores indicating higher fidelity. We adopt the FGD evaluation pipeline provided by the BEAT dataset authors, which includes a pretrained autoencoder used consistently across several recent gesture generation studies [40].

For baseline comparisons, we report FGD scores as published in prior work, without rerunning the evaluations ourselves. These scores are taken from papers that used the same dataset and the official BEAT evaluation model, including GestureDiffuCLIP [7], CaMN [40], and Semantic Gesticulator [54]. This ensures a consistent and fair basis of comparison across methods.

We then evaluate ConGRets and its ablations using the same autoencoder and protocol:

- **Encoded ConGRets** (264.8): a direct end-to-end setup where the fused latent is decoded to motion without retrieval.
- **ConGRets-latent** (95.4): latent is matched to a cluster centroid; a gesture latent is sampled and then decoded.
- **ConGRets** (84.62): latent is matched to a cluster, and the corresponding real gesture unit is retrieved and replayed without decoding.

Among all systems, ConGRets achieves the lowest FGD (84.62)—outperforming both recent generative models such as GestureDiffuCLIP (85.17), Semantic Gesticulator (89.15), and baseline CaMN (123.7). The ablation results demonstrate a clear trend: decoding from latent space, even when cluster-guided, introduces reconstruction loss that impacts fidelity. In contrast, ConGRets preserves temporal coherence and motion realism by directly retrieving high-variance real motion units, bypassing the limitations of neural decoding.

This highlights a key design strength of ConGRets: by replacing motion synthesis with learned retrieval, it avoids the quality degradation often associated with generation pipelines, while remaining fast and scalable.

Table 2. Summary of ConGRets performance across core evaluation axes. SP = Speaker Style Accuracy, CC = Content–Cluster Accuracy, FGD = Fréchet Gesture Distance (lower is better). SP and CC are reported at a 30-minute style context.

System	Execution Time (CPU)	Memory Usage	FGD ↓	SP / CC Accuracy (30 min) ↑
Baselines				
Rule-Based (Ali et al.)	16s (50 queries)	>2GB	–	–
Speech2Gesture [21]	8s (50 queries)	–	–	–
TrimodalContext [50]	3s (50 queries)	–	–	71% / 64%
Seq2Seq [51]	–	–	261.3	–
CaMN [40]	–	–	123.7	–
GestureDiffuCLIP [7]	–	–	85.17	–
Semantic Gesticulator [54]	–	–	89.15	–
ConGRets (ours)	**<500ms (1000 queries)**	**100MB**	**84.62**	**99% / 89%**
Ablation				
ConGRets-latent (ours)	–	–	95.4	– / –
Encoded ConGRets (ours)	–	–	264.8	76% / 42%

5.3 Retrieval Accuracy and Style Matching

Table 1 compares retrieval accuracy across three variants: - ZS-MSTM [16] - ConGRets - Encoded ConGRets (our latent-decoding variant)

We report three metrics: - **SP**: style (speaker) match, - **CG**: content–gesture match, - **CC**: content–cluster match.

As context length increases (5 to 30 minutes), performance improves across all models, with ConGRets consistently outperforming alternatives. For instance, speaker-style retrieval improves from 82% to 99%, while content-gesture accuracy rises from 15% to 55%.

Encoded ConGRets preserves style accuracy (74%→92%) but underperforms on content, suggesting compression artifacts degrade gesture fidelity. This reinforces the strength of retrieval-based output.

5.4 Zero-Shot Speaker Style Adaptation

We assessed ConGRets' ability to adapt to unseen speakers using only motion data (no audio or text). Cosine similarity matrices for style embeddings (Fig 7) show that even 1 minute of motion suffices for speaker separation, though 10 minutes yields clearer stylistic boundaries.

Figure 8 compares model variants: - (a) speaker-specific models, - (b) shared model with style embedding (ConGRets), - (c) ZS-MSTM baseline using multimodal style inference.

While speaker-specific models perform best for known speakers, ConGRets achieves comparable performance using only animation-based embeddings, avoiding retraining. ZS-MSTM, reproduced from its paper, performs reasonably well but relies on audio, text, and motion input, whereas ConGRets requires only motion.

5.5 Summary of Results

- **ConGRets is fast and lightweight**, outperforming prior rule-based and end-to-end systems in execution time and memory usage.
- **Retrieval accuracy improves with speaker context**, and outperforms ZS-MSTM and compressed variants at all data levels.
- **Zero-shot adaptation generalizes well**, requiring only motion data to match or exceed baselines that depend on multimodal input.
- **Retrieving real gesture units is most effective**, yielding the best FGD scores and validating our design choice of decoupled generation and style-aware retrieval.

These results establish ConGRets as a fast, style-aware, and scalable framework for co-speech gesture generation, suitable for real-time interactive systems.

6 Discussion

This section reflects on the practical and scientific implications of ConGRets, evaluates its limitations, and outlines promising directions for future research.

6.1 Implications for Real-World Deployment

ConGRets demonstrates that decoupling gesture generation from gesture retrieval offers compelling advantages for practical, real-time systems. Unlike generative models that synthesize gestures frame-by-frame, often requiring expensive rendering and postprocessing, ConGRets retrieves pre-cleaned, reusable motion segments conditioned on text and speaker style. This makes it highly suitable for edge deployment, interactive chatbots, previsualization tools, and real-time game engines, where speed, consistency, and low computational overhead are essential.

Our retrieval-based design allows integration into existing animation pipelines, enabling developers to plug ConGRets into avatars or agents without re-engineering rendering or simulation backends. Unlike many end-to-end models which remain largely experimental or lack downstream integration examples, ConGRets can be deployed as a lightweight module without API dependencies or GPU requirements.

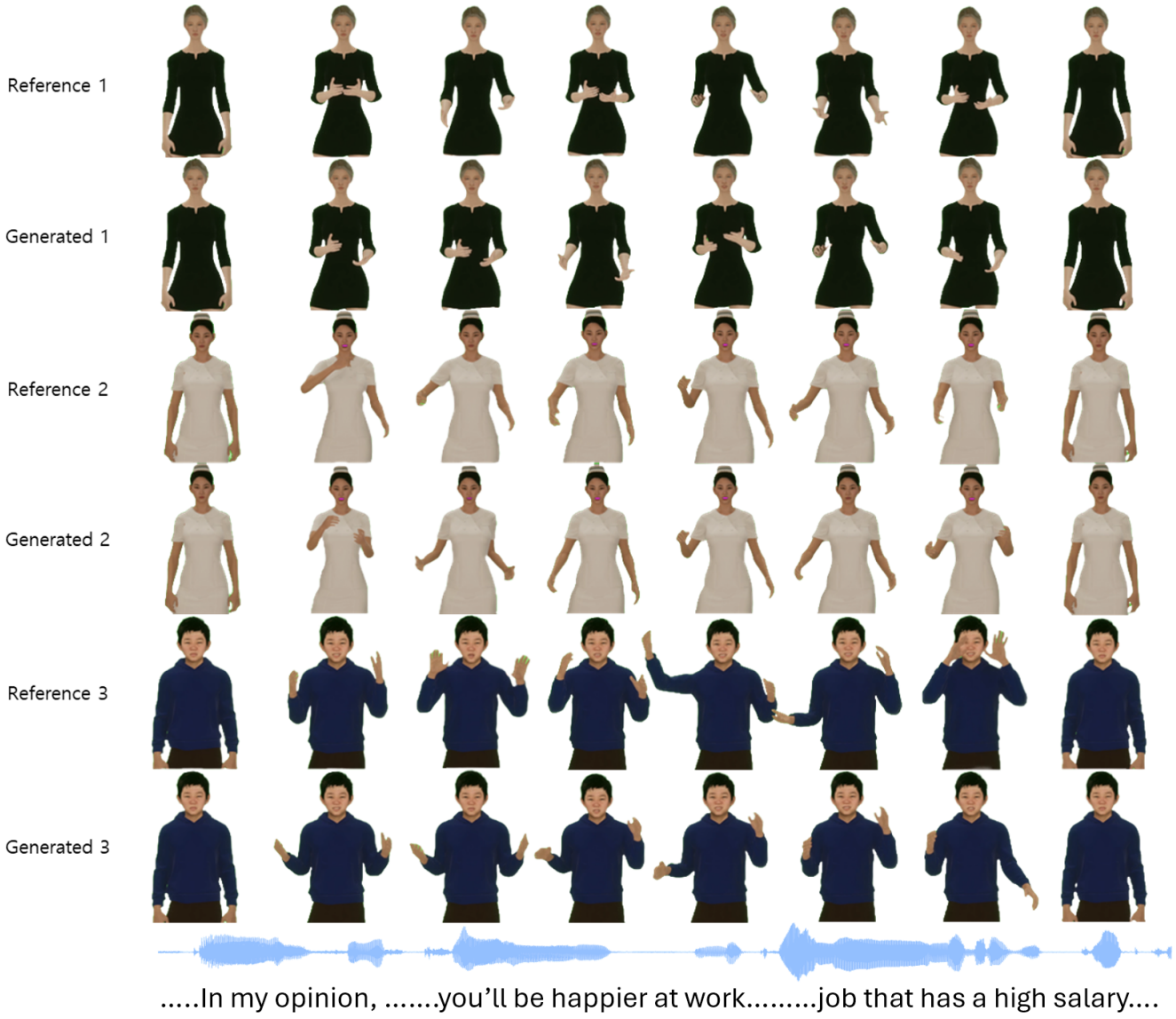


Fig. 9. Speaker style demonstration: showcasing three speakers using the same original utterance, with corresponding body animations provided as references. The generated gestures are produced by our system.

Furthermore, the system's ability to infer speaker-specific style from motion alone (requiring as little as 10 minutes of unlabeled 3D motion) facilitates fast personalization. This feature opens a pathway toward scalable style-conditioned gesture systems that can adapt to new users or characters without retraining, annotation, or multimodal input.

6.2 Strengths of the Retrieval Framework

Beyond performance, the retrieval mechanism supports a high degree of diversity and continuity. By segmenting long utterances into textual chunks and retrieving style-aligned gesture units, ConGRets ensures that gestures are temporally appropriate and stylistically consistent. Compared to traditional rule-based approaches, our framework offers wider variability without sacrificing runtime reliability.

Importantly, our design provides strong quality guarantees: because gesture units are preprocessed and manually validated, output animations are inherently retargeted and glitch-free. This contrasts with generative pipelines that often require additional constraints, denoising, or quality filtering steps to avoid implausible poses.

6.3 Limitations

While retrieval offers robustness and speed, it imposes limitations on generative flexibility. Unlike neural models that may combine motion primitives or interpolate novel variants, ConGRets is restricted to the distribution of preexisting gesture units. Rare or emergent motions cannot be synthesized unless explicitly present in the gesture library.

Additionally, gesture continuity across long utterances remains non-trivial. Although we implement blending at gesture boundaries and design gesture units with overlapping frame buffers, artifacts may still occur—especially during rapid topic shifts or in low-motion speakers. Future work could explore learned transition models or sequence-aware blending policies.

The speaker style encoder is intentionally speaker-specific and not generalized to abstract style categories (e.g., energetic vs. calm). While this supports high-fidelity personalization, it limits generalization across style types. Moreover, style inference in low-motion speakers may produce minimal or ambiguous outputs, which while arguably faithful might reduce engagement in expressive systems.

6.4 Evaluation Gaps and Metric Critique

Current evaluation methods offer only partial insight. Metrics like Fréchet Gesture Distance (FGD), though widely used, are fragile in the co-speech domain due to the inherently many-to-many nature of gesture–text mappings. High FGD may penalize valid yet diverse outputs and may not correlate with user perception. Similarly, cosine similarity for speaker embeddings measures stylistic clustering but not communicative effectiveness.

While user studies are a potential alternative, their reliability is limited by subjective expectations and task framing. In prior work[36, 53], even ground truth motions have received moderate appropriateness scores due to evaluators' overreliance on literal alignment. Long-term user-in-the-loop studies tracking engagement, style fatigue, or perceived expressiveness over time may offer more actionable evaluation.

6.5 Future Work

Several directions emerge from our findings:

- **Multimodal Style Vectors:** Though our design avoids prosody for latency reasons, future variants could explore optional speech or audio-derived embeddings, especially in offline settings where quality trumps speed.
- **Online Retrieval Adaptation:** Since gesture units are modular, future systems could dynamically expand or refine the retrieval library using usage logs or reinforcement objectives.
- **Generalizing Style Embeddings:** Extending the style encoder to disentangle speaker identity from stylistic traits (e.g., gesture energy, hand dominance, tempo) could enable more flexible transfer and clustering.
- **Beyond Gesture:** The retrieval strategy can generalize to other motion domains such as facial animation, body posture, or full-scene action generation where pre-verified segments can be reused for fast synthesis.
- **Evaluation Improvements:** Alternative metrics that incorporate semantic alignment, gesture saliency, or attention to discourse structure may better reflect communicative quality. Additionally, controlled long-term user studies could provide more reliable behavioral insight than snapshot surveys.

ConGRets presents a fast, scalable, and style-aware alternative to generative co-speech gesture models. Its retrieval-based architecture supports real-time use without sacrificing personalization or perceptual quality. While it trades off some generative flexibility, its robustness and deployability make it a strong candidate for embodied conversational agents, digital humans, and interactive avatars.

7 Conclusion

This paper introduced **ConGRets**, a scalable, low-latency framework for co-speech gesture generation that departs from conventional generative paradigms by leveraging contrastive learning and retrieval over pre-verified gesture units. Designed for fast and reliable deployment, ConGRets matches latent representations of text and speaker style directly to clusters of real gesture segments, enabling gesture synthesis without the computational overhead or fragility of frame-wise generation.

Our experiments demonstrate that ConGRets delivers substantial improvements over prior rule-based and end-to-end models in speed, memory efficiency, and speaker-specific style alignment. With inference times below 500ms on consumer hardware and personalization achievable from as little as 10 minutes of motion data, ConGRets is well-suited for interactive systems, including mobile agents, virtual assistants, and embedded avatars.

The model's modularity, decoupling gesture selection from gesture playback, offers robustness, reusability, and integration flexibility. Its motion-only style vector supports zero-shot speaker adaptation without retraining or multimodal dependencies, advancing the goal of scalable personalization in gesture synthesis.

While retrieval constrains generative novelty, our results show that style-aware matching to a rich gesture library yields both high perceptual quality and distributional authenticity. Limitations remain in handling ambiguous inputs or long-term continuity, but these challenges open new research directions in transition modeling, abstract style disentanglement, and user-centric evaluation.

In summary, ConGRets provides a practical and extensible foundation for real-time gesture generation. Its speed, modularity, and adaptability make it a compelling alternative to generative models for deploying naturalistic, personalized digital characters in real-world human-computer interaction.

References

[1] Amal Abdulrahman and Deborah Richards. 2022. Is Natural Necessary? Human Voice versus Synthetic Voice for Intelligent Virtual Agents. *Multimodal Technologies and Interaction* 6, 7 (2022). doi:10.3390/mti6070051

[2] Chaitanya Ahuja, Dong Won Lee, and Louis-Philippe Morency. 2022. Low-Resource Adaptation for Personalized Co-Speech Gesture Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20566–20576.

[3] Chaitanya Ahuja, Dong Won Lee, Yukiko I Nakano, and Louis-Philippe Morency. 2020. Style Transfer for Co-speech Gesture Animation: A Multi-speaker

Conditional-Mixture Approach. *arxiv.org* (7 2020), 248–265. doi:10.1007/978-3-030-58523-5_15

[4] Ghazanfar Ali and Jae-In Hwang. 2022. Improving Co-speech gesture rule-map generation via wild pose matching with gesture units. *SIGGRAPH Asia 2022 Posters* PartF168147-3 (12 2022), 1575–1585. doi:10.1145/3550082

[5] Ghazanfar Ali, Woojoo Kim, Muhammad Shahid Anwar, Jae-In Hwang, and Ahyoung Choi. 2023. Expanding Multilingual Co-Speech Interaction: The Impact of Enhanced Gesture Units in Text-to-Gesture Synthesis for Digital Humans. (2023).

[6] Tenglong Ao, Qingzhe Gao, Yuke Lou, Baoquan Chen, and Libin Liu. 2022. Rhythmic Gesticulator: Rhythm-Aware Co-Speech Gesture Synthesis with Hierarchical Neural Embeddings. *ACM Trans. Graph.* 41, 6, Article 209 (nov 2022), 19 pages. doi:10.1145/3550454.3555435

[7] Tenglong Ao, Zeyi Zhang, and Libin Liu. [n. d.]. GestureDiffuCLIP: Gesture Diffusion Model with CLIP Latents. *ACM Trans. Graph.* ([n. d.]), 18 pages. doi:10.1145/3592097

[8] Eiichi Asakawa, Naoshi Kaneko, Dai Hasegawa, and Shinichi Shirakawa. 2022. Evaluation of text-to-gesture generation model using convolutional neural network. *Neural Networks* 151 (2022), 365–375.

[9] Uttaran Bhattacharya, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha. 2021. Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In *2021 IEEE virtual reality and 3D user interfaces (VR)*. IEEE, 1–10.

[10] Justine Cassell. 2000. Embodied conversational interface agents. *Commun. ACM* 43, 4 (2000), 70–78.

[11] Justine Cassell, Catherine Pelachaud, Norman Badler, Mark Steedman, Brett Achorn, Tripp Becket, Brett Douville, Scott Prevost, and Matthew Stone. 1994. Animated conversation: rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. 413–420.

[12] Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. 2000. *Embodied conversational agents*. MIT press.

[13] Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. 2001. Beat: the behavior expression animation toolkit. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 477–486.

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv preprint arXiv:2002.05709* (2020).

[15] Chung-Cheng Chiu, Louis-Philippe Morency, and Stacy Marsella. 2015. Predicting co-verbal gestures: A deep and temporal modeling approach. In *Intelligent Virtual Agents: 15th International Conference, IVA 2015, Delft, The Netherlands, August 26-28, 2015, Proceedings 15*. Springer, 152–166.

[16] Mireille Fares, Catherine Pelachaud, and Nicolas Obin. 2023. Zero-Shot Style Transfer for Multimodal Data-Driven Gesture Synthesis. In *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*. 1–4. doi:10.1109/FG57933.2023.10042658

[17] Ylva Ferstl, Michael Neff, and Rachel McDonnell. 2019. Multi-objective adversarial gesture generation. *Proceedings - MIG 2019: ACM Conference on Motion, Interaction, and Games* (2019). doi:10.1145/3359566.3360053

[18] Ylva Ferstl, Michael Neff, and Rachel McDonnell. 2021. ExpressGesture: Expressive Gesture Generation from Speech through Database Matching. *Computer Animation and Virtual Worlds* 32, 3-4 (2021), e2016. doi:10.1002/cav.2016

[19] Nan Gao, Zeyu Zhao, Zhi Zeng, Shuwu Zhang, Dongdong Weng, and Yihua Bao. 2024. GesGPT: Speech Gesture Synthesis With Text Parsing from ChatGPT. *IEEE Robotics and Automation Letters* 9, 3 (2024), 2718–2725. doi:10.1109/LRA.2023.3241801

[20] Saeed Ghorbani, Ylva Ferstl, Daniel Holden, Nikolaus F. Troje, and Marc-André Carbonneau. 2023. ZeroEGGS: Zero-Shot Example-Based Gesture Generation from Speech. *Computer Graphics Forum* 42, 2 (2023), 131–144. doi:10.1111/cgf.14734

[21] Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik. 2019. Learning Individual Styles of Conversational Gesture. (jun 2019). arXiv:1906.04160 http://arxiv.org/abs/1906.04160

[22] Jonathan Gratch, David DeVault, Gale M Lucas, and Stacy Marsella. 2015. Negotiation as a challenge problem for virtual humans. In *Intelligent Virtual Agents: 15th International Conference, IVA 2015, Delft, The Netherlands, August 26-28, 2015, Proceedings 15*. Springer, 201–215.

[23] Ikhsanul Habibie, Mohamed Elgharib, Kripasindhu Sarkar, Ahsan Abdullah, Simbarashe Nyatsanga, Michael Neff, and Christian Theobalt. 2022. A Motion Matching-Based Framework for Controllable Gesture Synthesis from Speech. In *ACM SIGGRAPH Conference Proceedings*. 1–9. doi:10.1145/3528233.3530750

[24] Arno Hartholt, David Traum, Stacy C Marsella, Ari Shapiro, Giota Stratou, Anton Leuski, Louis-Philippe Morency, and Jonathan Gratch. 2013. All together now: Introducing the virtual human toolkit. In *Intelligent Virtual Agents: 13th International Conference, IVA 2013, Edinburgh, UK, August 29-31, 2013. Proceedings 13*. Springer, 368–381.

[25] Hanseob Kim, Ghazanfar Ali, Bin Han, Hwang Youn Kim, Jieun Kim, Hyemin Shin, Gerard Jounghyun Kim, and Jae-In Hwang. 2024. ASAP for multi-outputs: auto-generating storyboard and pre-visualization with virtual actors based on screenplay. *Multimedia Tools and Applications* (2024), 1–24.

[26] Hanseob Kim, Ghazanfar Ali, Seungwon Kim, Gerard J Kim, and Jae-In Hwang. 2021. Auto-generating Virtual Human Behavior by Understanding User Contexts. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 591–592.

[27] Hanseob Kim, Ghazanfar Ali, Andréas Pastor, Myungho Lee, Gerard J Kim, and Jae-In Hwang. 2021. Silhouettes from Real Objects Enable Realistic Interactions with a Virtual Human in Mobile Augmented Reality. *Applied Sciences* 11, 6 (2021), 2763.

[28] Hanseob Kim, Bin Han, Jieun Kim, Muhammad Firdaus Syawaludin Lubis, Gerard Jounghyun Kim, and Jae-In Hwang. 2024. Engaged and Affective Virtual Agents: Their Impact on Social Presence, Trustworthiness, and Decision-Making in the Group Discussion. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 149, 17 pages. doi:10.1145/3613904.3642917

[29] Hanseob Kim, Jieun Kim, Ghazanfar Ali, and Jae-In Hwang. 2022. No-code Digital Human for Conversational Behavior. In *SIGGRAPH Asia 2022 Posters*. 1–2.

[30] Michael Kipp. 2005. *Gesture generation by imitation: From human behavior to computer character animation*. Universal-Publishers.

[31] Michael Kipp, Michael Neff, Kerstin H Kipp, and Irene Albrecht. 2007. Towards natural gesture synthesis: Evaluating gesture units in a data-driven approach to gesture synthesis. In *IVA*, Vol. 7. 15–28.

[32] Stefan Kopp, Brigitte Krenn, Stacy Marsella, Andrew N Marshall, Catherine Pelachaud, Hannes Pirker, et al. 2004. Towards a common framework for multimodal generation: The behavior markup language. In *International conference on intelligent virtual agents*. Springer, Berlin, Heidelberg, 205–217.

[33] Stefan Kopp, Brigitte Krenn, Stacy Marsella, Andrew N Marshall, Catherine Pelachaud, Hannes Pirker, Kristinn R Thórisson, and Hannes Vilhjálmsson. 2006. Towards a common framework for multimodal generation: The behavior markup language. In *Intelligent Virtual Agents: 6th International Conference, IVA 2006, Marina Del Rey, CA, USA, August 21-23, 2006. Proceedings 6*. Springer, 205–217.

[34] Taras Kucherenko, Dai Hasegawa, Gustav Eje Henter, Naoshi Kaneko, and Hedvig Kjellström. 2019. Analyzing input and output representations for speech-driven gesture generation. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*. 97–104.

[35] Taras Kucherenko, Patrik Jonell, and Sanne Van Waveren. 2020. Gustav Eje Henter, Simon Alexandersson, Iolanda Leite, and Hedvig Kjellström. 2020. Gesticulator: A framework for semantically-aware speech-driven gesture generation. In *Proceedings of the 2020 international conference on multimodal interaction*. 242–250.

[36] Taras Kucherenko, Patrik Jonell, Youngwoo Yoon, Pieter Wolfert, and Gustav Eje Henter. 2021. A Large, Crowdsourced Evaluation of Gesture Generation Systems on Common Data: The GENEA Challenge 2020. *International Conference on Intelligent User Interfaces, Proceedings IUI* (4 2021), 11–21. doi:10.1145/3397481.3450692

[37] Gilwoo Lee, Zhiwei Deng, Shugao Ma, Takaaki Shiratori, Siddhartha Srinivasa, and Yaser Sheikh. 2019. Talking With Hands 16.2M: A Large-Scale Dataset of Synchronized Body-Finger Motion and Audio for Conversational Motion Analysis and Synthesis. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 763–772. doi:10.1109/ICCV.2019.00085

[38] Jina Lee and Stacy Marsella. 2006. Nonverbal behavior generator for embodied conversational agents. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4133 LNAI (2006), 243–255. doi:10.1007/11821830_20

[39] Margot Lhommet, Yuyu Xu, and Stacy Marsella. 2015. Cerebella: automatic generation of nonverbal behavior for virtual humans. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[40] Haiyang Liu, Zihao Zhu, Naoya Iwamoto, Yichen Peng, Zhengqing Li, You Zhou, Elif Bozkurt, and Bo Zheng. 2022. BEAT: A Large-Scale Semantic and Emotional Multi-modal Dataset for Conversational Gestures Synthesis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 13667 LNCS (2022), 612–630. doi:10.1007/978-3-031-20071-7_36/TABLES/6

[41] Stacy Marsella, Yuyu Xu, Margaux Lhommet, Andrew Feng, Stefan Scherer, and Ari Shapiro. 2013. Virtual character performance from speech. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 25–35.

[42] Michael Neff, Michael Kipp, Irene Albrecht, and Hans-Peter Seidel. 2008. Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Transactions on Graphics (TOG)* 27, 1 (2008), 1–24.

[43] Simbarashe Nyatsanga, Taras Kucherenko, Chaitanya Ahuja, Gustav Eje Henter, and Michael Neff. 2023. A Comprehensive Review of Data-Driven Co-Speech

Gesture Generation. 42 (1 2023). Issue 2. https://arxiv.org/abs/2301.05339v3

[44] Brian Ravenet, Catherine Pelachaud, Chloé Clavel, and Stacy Marsella. 2018. Automating the production of communicative gestures in embodied characters. *Frontiers in psychology* 9 (2018), 1144.

[45] Manuel Rebol, Christian Gütl, and Krzysztof Pietroszek. 2021. Real-time gesture animation generation from speech for virtual human interaction. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–4.

[46] Marcus Thiebaux, Stacy Marsella, Andrew N Marshall, and Marcelo Kallmann. 2008. Smartbody: Behavior realization for embodied conversational agents. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. 151–158.

[47] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. *arXiv preprint arXiv:1908.08962v2* (2019).

[48] Pieter Wolfert, Nicole Robinson, and Tony Belpaeme. 2022. A review of evaluation practices of gesture generation in embodied conversational agents. *IEEE Transactions on Human-Machine Systems* (2022).

[49] Yanzhe Yang, Jimei Yang, and Jessica Hodgins. 2020. Statistics-based Motion Synthesis for Social Conversations. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 201–212.

[50] Youngwoo Yoon, Bok Cha, Joo-Haeng Lee, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. 2020. Speech Gesture Generation from the Trimodal Context of Text, Audio, and Speaker Identity. *ACM Transactions on Graphics* 39, 6 (2020).

[51] Youngwoo Yoon, Woo-Ri Ko, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. 2019. Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 4303–4309.

[52] Youngwoo Yoon, Keunwoo Park, Minsu Jang, Jaehong Kim, and Geehyuk Lee. 2021. Sgtoolkit: An interactive gesture authoring toolkit for embodied conversational agents. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 826–840.

[53] Youngwoo Yoon, Pieter Wolfert, Taras Kucherenko, Carla Viegas, Teodor Nikolov, Mihail Tsakov, and Gustav Eje Henter. 2022. The GENEA Challenge 2022: A large evaluation of data-driven co-speech gesture generation. *ACM International Conference Proceeding Series* (11 2022), 736–747. doi:10.1145/3536221.3558058

[54] Zeyi Zhang, Tenglong Ao, Yuyao Zhang, Qingzhe Gao, Chuan Lin, Baoquan Chen, and Libin Liu. 2024. Semantic Gesticulator: Semantics-Aware Co-Speech Gesture Synthesis. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–17.

[55] Chi Zhou, Tengyue Bian, and Kang Chen. 2022. GestureMaster: Graph-Based Speech-Driven Gesture Generation. In *Proceedings of the 2022 International Conference on Multimodal Interaction* (Bengaluru, India) *(ICMI '22)*. Association for Computing Machinery, New York, NY, USA, 764–770. doi:10.1145/3536221.3558063

[56] Lingting Zhu, Xian Liu, Xuanyu Liu, Rui Qian, Ziwei Liu, and Lequan Yu. 2023. Taming Diffusion Models for Audio-Driven Co-Speech Gesture Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10544–10553.